Лабораторная работа №9

Архитектура компьютера

Овчинников Антон

Содержание

# 1 Цель работы

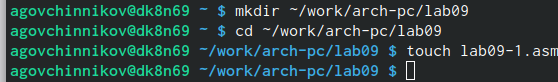
Приобрести навыки написания программ с использованием подпрограмм. Знакомство с методами откладки при помощи GDB и его основными возможностями

# 2 Теоретическое введение

Отладка — это процесс поиска и исправления ошибок в программе. В общем случае его можно разделить на четыре этапа: • обнаружение ошибки; • поиск её местонахождения; • определение причины ошибки; • исправление ошибки. Можно выделить следующие типы ошибок: • синтаксические ошибки — обнаруживаются во время трансляции исходного кода и вызваны нарушением ожидаемой формы или структуры языка; • семантические ошибки — являются логическими и приводят к тому, что программа запускается, отрабатывает, но не даёт желаемого результата; • ошибки в процессе выполнения — не обнаруживаются при трансляции и вызывают пре- рывание выполнения программы (например, это ошибки, связанные с переполнением или делением на ноль). Второй этап — поиск местонахождения ошибки. Некоторые ошибки обнаружить доволь- но трудно. Лучший способ найти место в программе, где находится ошибка, это разбить программу на части и произвести их отладку отдельно друг от друга. Третий этап — выяснение причины ошибки. После определения местонахождения ошибки обычно проще определить причину неправильной работы программы. Последний этап — исправление ошибки. После этого при повторном запуске программы, может обнаружиться следующая ошибка, и процесс отладки начнётся заново.

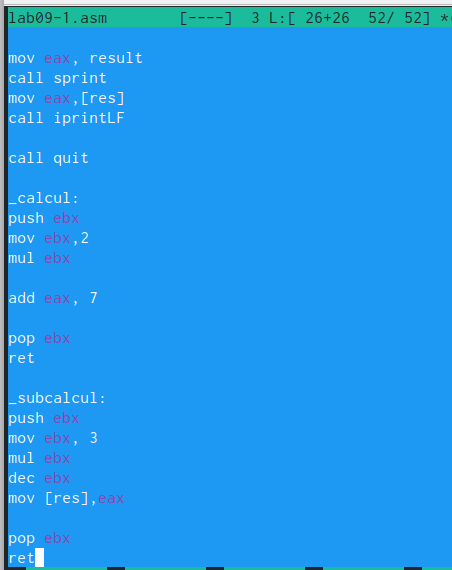
# 3 Выполнение лабораторной работы

Создаю рабочую директорию и файл. (рис. ??).



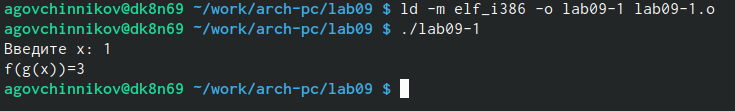
Создание файла

Записываю туда программу из листинга,при этом исправив опечатки. (рис. ??).



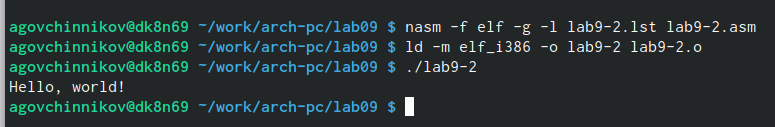
Текст файла

Проверим работу файла (рис. ??).



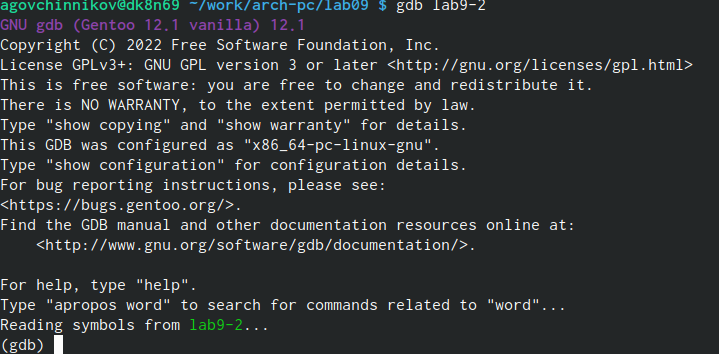
Вывод программы

Создаю файл lab9-2.asm и проассемблируем его с другими ключами, чтобы была возможность открыть этот файл через gdb. (рис. ??).



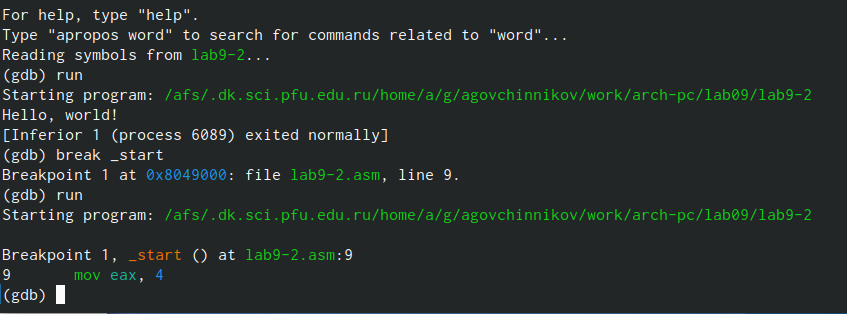
Работа программы

Открываю файл lab9-2 с помощью gdb (рис. ??).



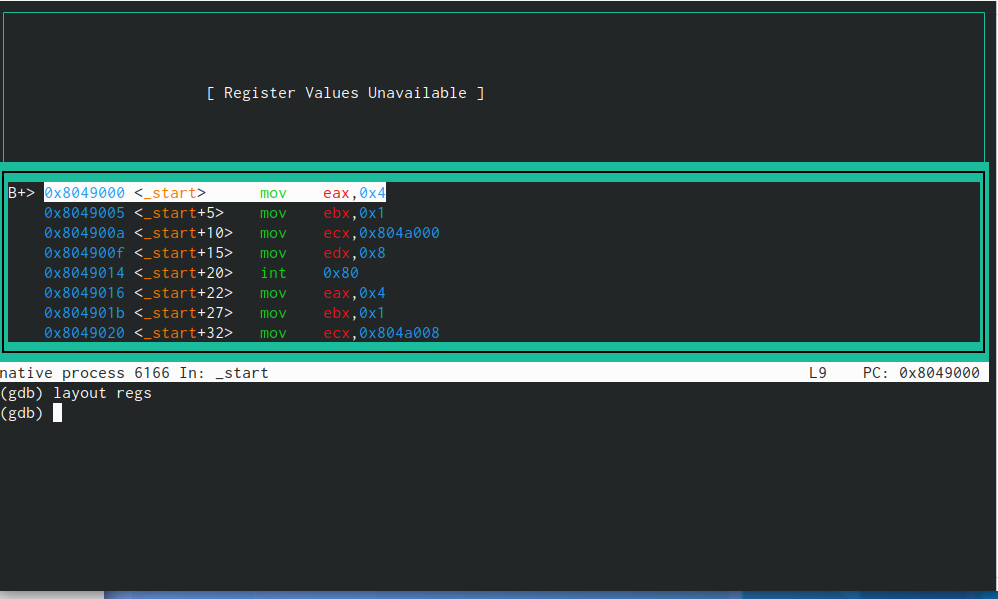
Запускаем файл в gdb

Поставим точку остановки на метке \_start (рис. ??).



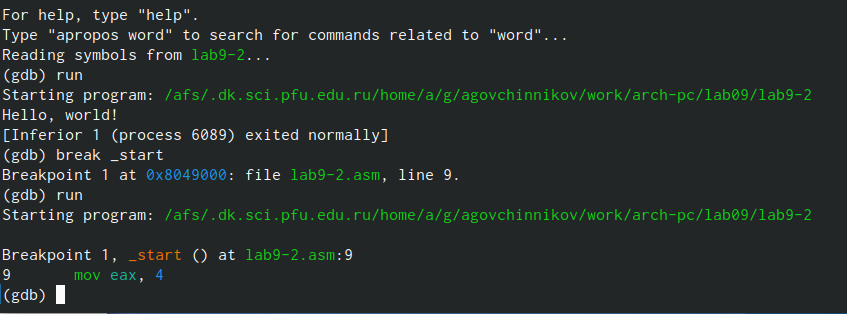
breakpoint

В представлении АТТ в виде 16-ричного числа записаны первые аргументы всех комманд, а в представлении intel так записываются адреса вторых аргумантов. Включим режим псевдографики, с помощью которго отбражается код программы и содержимое регистров (рис. ??).



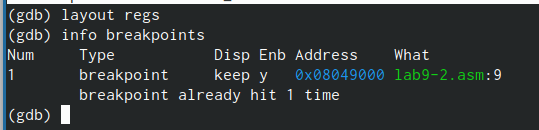
Содержимое регистров

Поставим точку остановки на метке \_start (рис. ??).



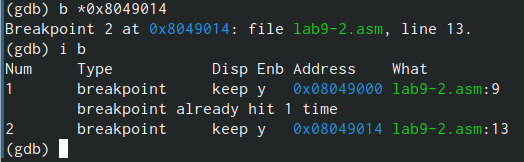
breakpoint

Установим еще одну точку останова (рис. ??).



Установка точки останова

Посмотрим информацию о наших точках останова. Сделать это можно коротко командой i b (рис. ??).



Информация о точках останова

В откладчике можно вывести текущее значение переменных(рис. ??).

Вывод по имени

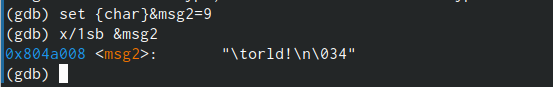
Вывод по имени

В откладчике можно вывести текущее значение переменных(рис. ??).

Вывод по имени

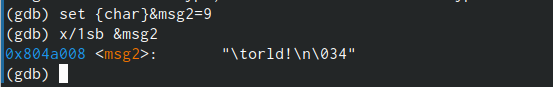
Вывод по имени

Также можно обращаться по адресам переменных. Здесь был заменен первый символ переменной msg2 на символ отступа(рис. ??).



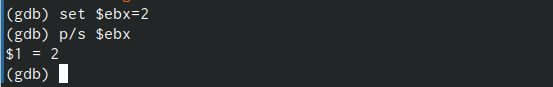
Код программы

Также можно обращаться по адресам переменных. Здесь был заменен первый символ переменной msg2 на символ отступа(рис. ??).



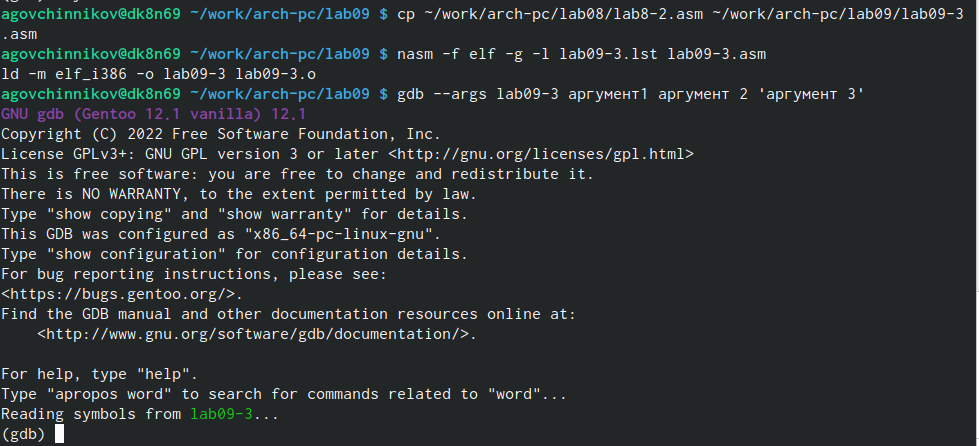
Код программы

Зададим регистру значения (рис. ??).



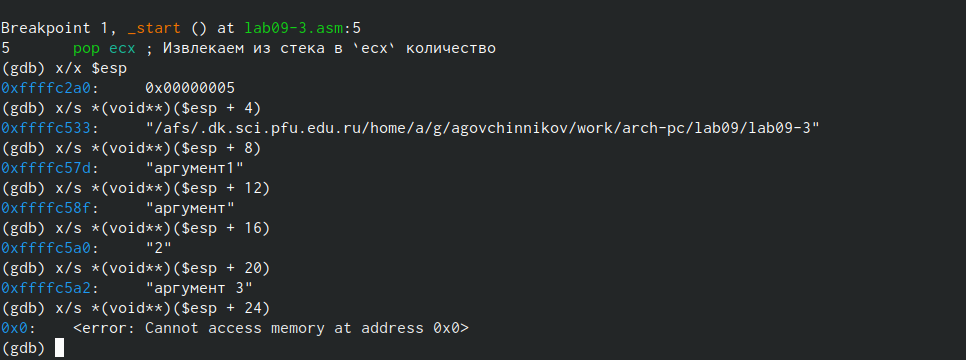
Значение регистра

Скопируем файл из предыдущей лабораторной, переименуею и создадим исполняемый файл. Создадим точку останова на метке \_start и запустим программу.(рис. ??).



Запуск программы

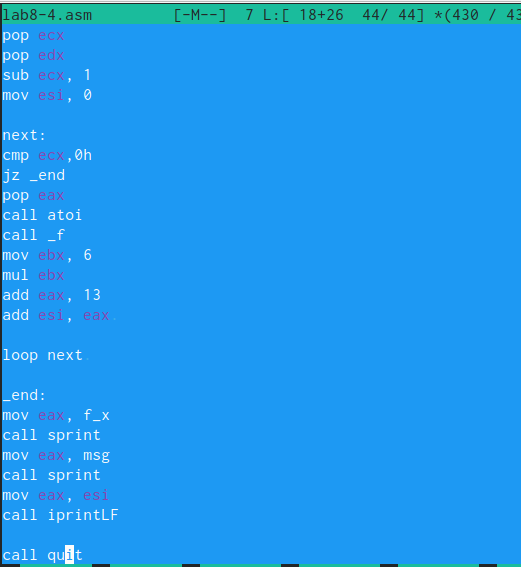
Посмотрим на все остальные аргументы в стеке. Их адреса располагаются в 4 байтах друг от друга. (рис. ??)



Адреса аргументов

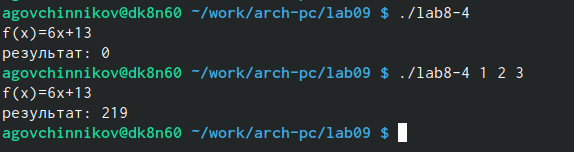
# 4 Задания для самостотельной работы

Программа из лабороторной 9, но с использованием подпрограмм. (рис. ??)



Текст листинга

Запускаю исполняемый файл. (рис. ??)



Результат программы

# 5 Выводы

Я приобрел навыки написания программ с использованием подпрограмм. Я ознакомился с методами откладки при помощи GDB и его основными возможностями.